

### AMENDMENTS TO THE CLAIMS

The following listing of claims replaces all prior versions, and listings, of claims in the captioned patent application:

1. (Currently Amended) A computer system comprising:

at least one first node connected to at least one second node, wherein said at least one first node and said at least one second node have ~~having~~ at least one first memory and at least one second memory, respectively; and

a ~~first-threadlet~~ configured to cause a ~~for causing a first~~ program to run in said computer system executed by said at least one first node when said at least one first memory is local to said ~~first-threadlet~~; and

a parcel for moving said threadlet from said at least one first memory to said at least one second memory to execute said threadlet by said at least one second node when said at least one second memory is local to said threadlet.

2. (Currently Amended) The computer system of claim 1, wherein said ~~first~~ program requires access to a first memory location to run.

3. (Currently Amended) The computer system of claim 2, claim 1, ~~claim 1,~~ wherein said ~~first~~ threadlet is capable of determining that whether or not said first memory location is local to said ~~first~~ threadlet.

4. (Currently Amended) The computer system of claim 1, wherein said computer system is capable of saving said ~~first~~ threadlet and a state of said ~~first~~ threadlet in said parcel, ~~a parcel~~.

5. (Currently Amended) The computer system of claim 4, wherein said ~~first threadlet~~ state includes a ~~first~~ program counter that references said ~~first~~ program.

6. (Original) The computer system of claim 5, wherein said computer system is capable of injecting said parcel into a communication network.

7. (Currently Amended) The computer system of claim 6, wherein said computer system is capable of determining that said parcel has reached said second ~~a second~~ node after being injected into said communication network.

8. (Currently Amended) The computer system of claim 7, wherein said computer system is capable of unpacking said ~~first threadlet~~ state from said parcel and restarting said ~~first~~ threadlet in said ~~first threadlet~~ state when said ~~first~~ threadlet reaches said second node.

9. (Currently Amended) The computer system of claim 2, ~~claim 8~~, wherein said computer system is capable of verifying that said first memory location is within said second node.

10. (Currently Amended) The computer system of claim 1, wherein said ~~first~~ threadlet includes said ~~first~~ program.

11. (Currently Amended) The computer system of claim 1, wherein said computer system includes a plurality of said first nodes and each first memory of each of said first nodes includes a copy of said ~~first~~ program.

12. (Currently Amended) The computer system of claim 1, wherein said ~~first~~ program comprising an additional threadlet for causing said program to run in said computer system executed by said at least one first node when said at least one first memory is local to said additional threadlet; and wherein said parcel moves said additional threadlet from said at least one first memory to said at least one second memory to execute said additional threadlet by said at least one second node when said at least one second memory is local to said additional threadlet. ~~causes a second threadlet to be created, and wherein said second threadlet causes a second program to run in said computer system when at least one second memory of a second node is local to said first threadlet.~~

13. (Currently Amended) The computer system of claim 1, wherein a minimum state of said ~~first~~ threadlet is a wide word.

14. (Original) The computer system of claim 1, wherein said threadlet has an extended state of full word plus a wide word.

15. (Original) The computer system of claim 1, wherein said first node is on a PIM-enhanced memory chip.

16. (Original) The computer system of claim 1, wherein said at least one first node is a plurality of first nodes.

17. (Original) The computer system of claim 16, wherein each of said first nodes is on a PIM-enhanced memory chip.

18. (Original) The computer system of claim 16, wherein each of said first nodes is on a set of PIM-enhanced memory chips interconnected by a communication network.

19. (Currently Amended) A method comprising the steps of:

executing a program having a threadlet;  
executing said threadlet by a first node when a target memory required to execute said program at said first node is local to said threadlet; and  
moving said threadlet to a second node when said target memory required to execute said program at said second node is local to said threadlet.  
~~providing a first threadlet for causing a first program to run in a computer system when at least one first memory of at least one first node of said computer system is local to said first threadlet; and~~  
~~said first threadlet executing said first program in said first node.~~

20. (Canceled) The method of claim 19, wherein said first program requires access to a first memory location to run.

21. (Currently Amended) The method of claim 19, wherein said ~~first threadlet~~ is capable of determining whether or not said target memory ~~first memory location~~ is local to said first threadlet.

22. (Currently Amended) The method of claim 19, further comprising saving said ~~first~~ threadlet and a state of said ~~first~~-threadlet ~~is in~~ a parcel.

23. (Currently Amended) The method of claim 22, wherein said ~~first threadlet~~-state includes a ~~first~~ program counter that references said ~~first~~-program.

24. (Currently Amended) The method of claim 22, ~~claim 23~~, further comprising injecting said parcel into a communication network.

25. (Currently Amended) The method of claim 24, wherein said computer system determines if said parcel has reached ~~a second~~ said second node after being injected into said communication network.

26. (Currently Amended) The method of claim 25, further unpacking said ~~first threadlet~~-state from said parcel and restarting said ~~first~~-threadlet in said ~~first threadlet~~ state when said ~~first~~ threadlet reaches said second node.

27. (Currently Amended) The method of claim 26, further comprising verifying that said target memory ~~first memory location~~ is within said second node.

28. (Currently Amended) The method of claim 19, wherein said ~~first~~-threadlet includes said ~~first~~-program.

29. (Canceled) The method of claim 19, wherein said computer system includes a plurality of said first nodes and each first memory of each of said first nodes includes a copy of said first program.

30. (Currently Amended) The method of claim 19, wherein said ~~first~~-program comprises an additional ~~causes a second threadlet to be created, and wherein said second threadlet causes a second program to run in said computer system when at least one second memory of a second node is local to said first threadlet.~~

31. (Currently Amended) The method of claim 19, wherein a minimum state of said ~~first~~ threadlet is a wide word.

32. (Original) The method of claim 19, wherein said threadlet has an extended state of full word plus a wide word.

33. (Original) The method of claim 19, wherein said first node is on a PIM-enhanced memory chip.

34. (Original) The method of claim 19, wherein said at least one first node is a plurality of first nodes.

35. (Original) The method of claim 34, wherein each of said first nodes is on a PIM-enhanced memory chip.

36. (Original) The method of claim 34, wherein each of said first nodes is on a set of PIM-enhanced memory chips interconnected by a communication network.

37. (New) The method of claim 19, further comprising suspending said threadlet at said second node when said threadlet determines that said threadlet is not the next operation for said program.

38. (New) The method of claim 30, further comprising the steps of: executing said additional threadlet by said first node when said target memory required to execute said program at said first node is local to said additional threadlet; and moving said threadlet to said second node when said target memory required to execute said program at said second node is local to said additional threadlet.